KubeStellar

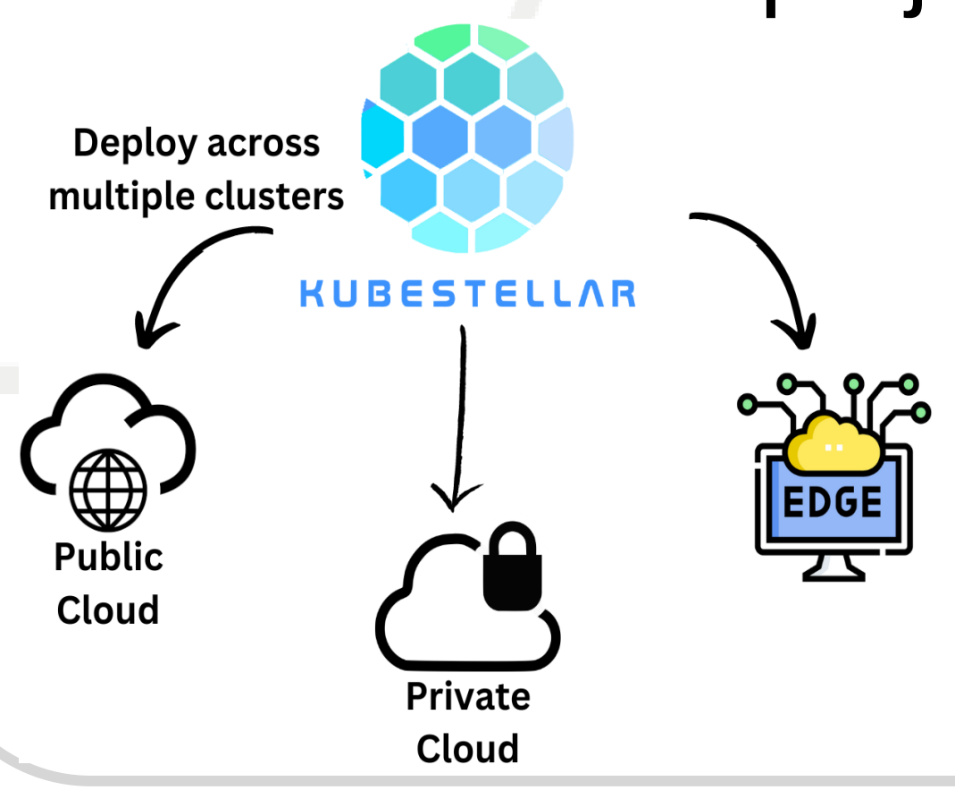
Students: Rickie Chen, Aishwarya Mathew
Mentors: Andy Anderson

Georgia Tech
Open Source Program Office

Project Overview

Kubernetes clusters are groups of servers that work together to run applications. They are used to manage and scale containerized applications.

KubeStellar is a project by the Cloud Native Computing Foundation (CNCF) designed to make it easier to deploy and manage applications on multiple Kubernetes clusters.



Goals and Milestones

- Installing a Kubernetes Distribution and learning its uses
- Optimizing the workspace by using autocomplete/aliases
- Deploy game 2048 yaml via cluster using Kubernetes distribution
- Installing helm and ingress nginx controller via helm
- Installing KubeStellar helm core chart and using it deploy game 2048
- KubeStellar installation script

Open Source Outcomes

- Opened more than 30 issues related to bugs/features
- Raise more than 10 PRs to repair documentation, code, installation, etc.
- Published 8 blog posts on medium, where 1 is a YouTube tutorial on how to deploy game-2048 onto two different remote clusters
- Used our experiences to confirm & challenge assumptions made by project maintainers
- Found mechanical and functional errors in KubeStellar's documentation and installation scripts
- Enhanced documentation by removing redundant and confusing content, which should help with project adoption
- Reinforced quality checks like the need for continuous spelling checks

Highlights and Accomplishments – SubProject 1

Aim: The goal is to install Kubernetes and create a remote cluster on which the game called '2048' is to be deployed. Additionally, collect feedback on the process of setting up the environment and using Kubernetes-native deployment tools like kubectl, helm, kind.

Installations

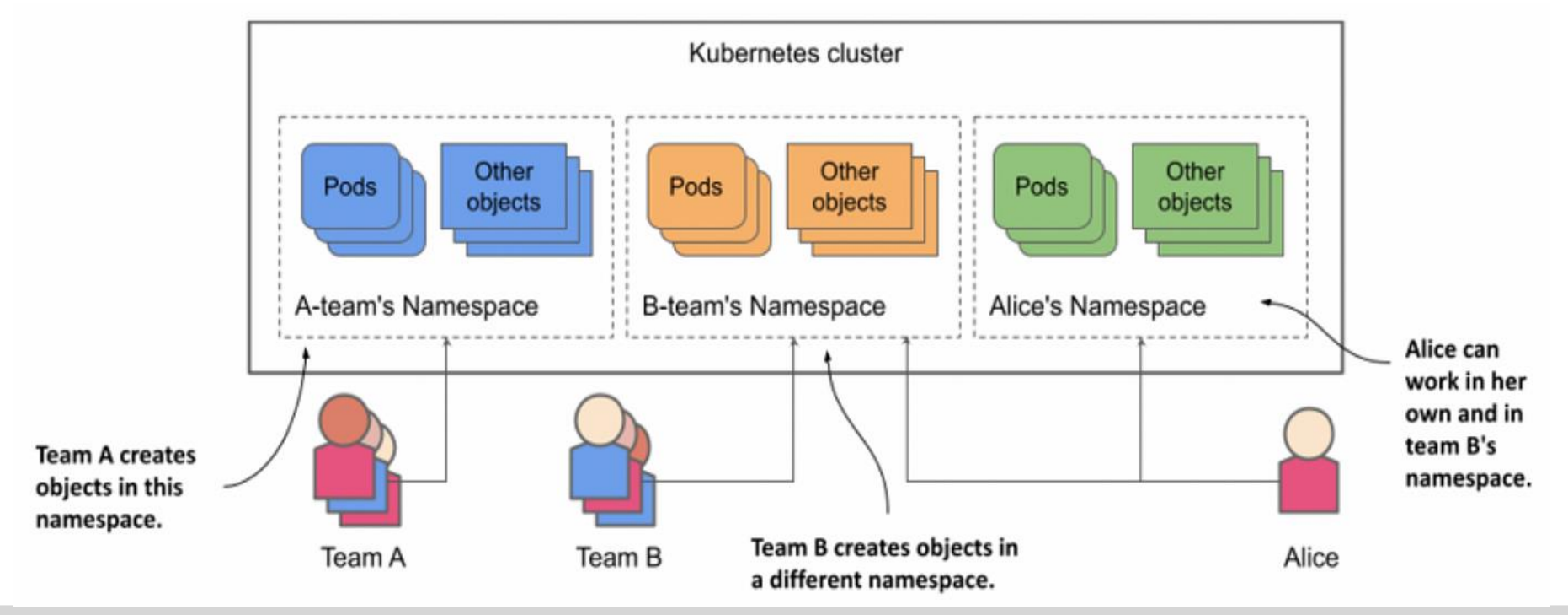
- Go lang
- Docker
- Kind, Minikube
- Kubectl, Kubectx
- Helm
- Kubernetes

Productivity:

- Aliases
- OhMyZsh terminal & powerlevel10k theme, fonts, context display

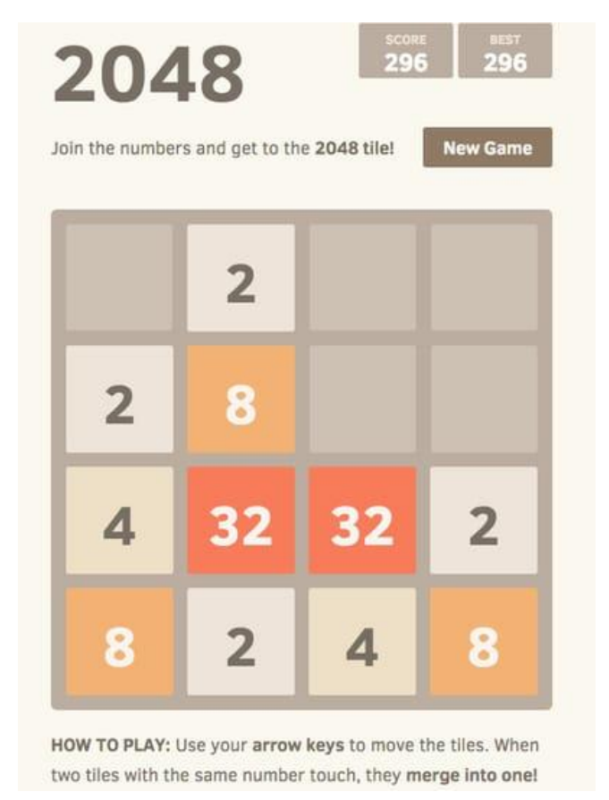
Steps

- Used kind to create a remote cluster
- Created a yaml file for Game2048
- Used kubectl to deploy this:
  `kubectl apply -f deployment.yml`
- Open the URL where the app is running:
  `minikube service service-2048 -n game-2048`


Kubernetes cluster
Pods
Other objects
A-team's Namespace
Pods
Other objects
B-team's Namespace
Pods
Other objects
Alice's Namespace
Alice can work in her own and in team B's namespace.
Team A creates objects in this namespace.
Team A
Team B
Team B creates objects in a different namespace.
Alice

Kubernetes Cluster

- **Namespace**: Isolates resources for the game.
- **Deployment**: Manages 5 instances of the 2048 application.
- **Service**: Exposes the application within the cluster and on a NodePort.
- **Ingress**: Provides external access to the application like HTTP or HTTPS. It can also provide load balancing.

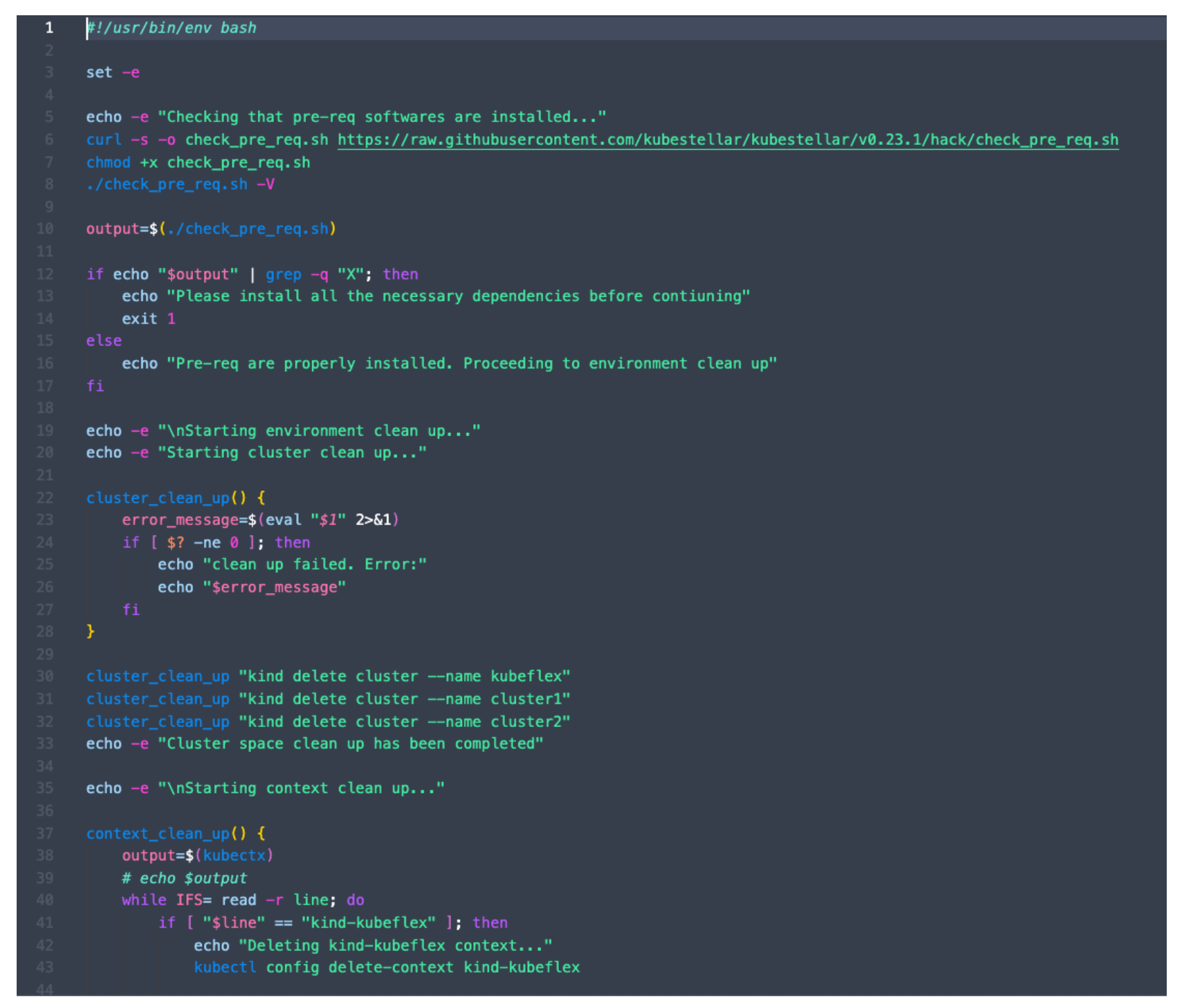


Highlights and Accomplishments – SubProject 2

Aim: The goal is to document the usage of KubeStellar helm core chart to deploy game 2048 onto 2 different remote cluster. We also want to create a script that creates 2 remote clusters seamlessly.

Steps

- Create 1 cluster called kind-kubeflex and 2 remote called cluster1 and cluster2
- Apply the deployment file onto the kind-kubeflex
- Check to see if the pods are ready on both clusters by switching context
- Open the link to both clusters in a browser

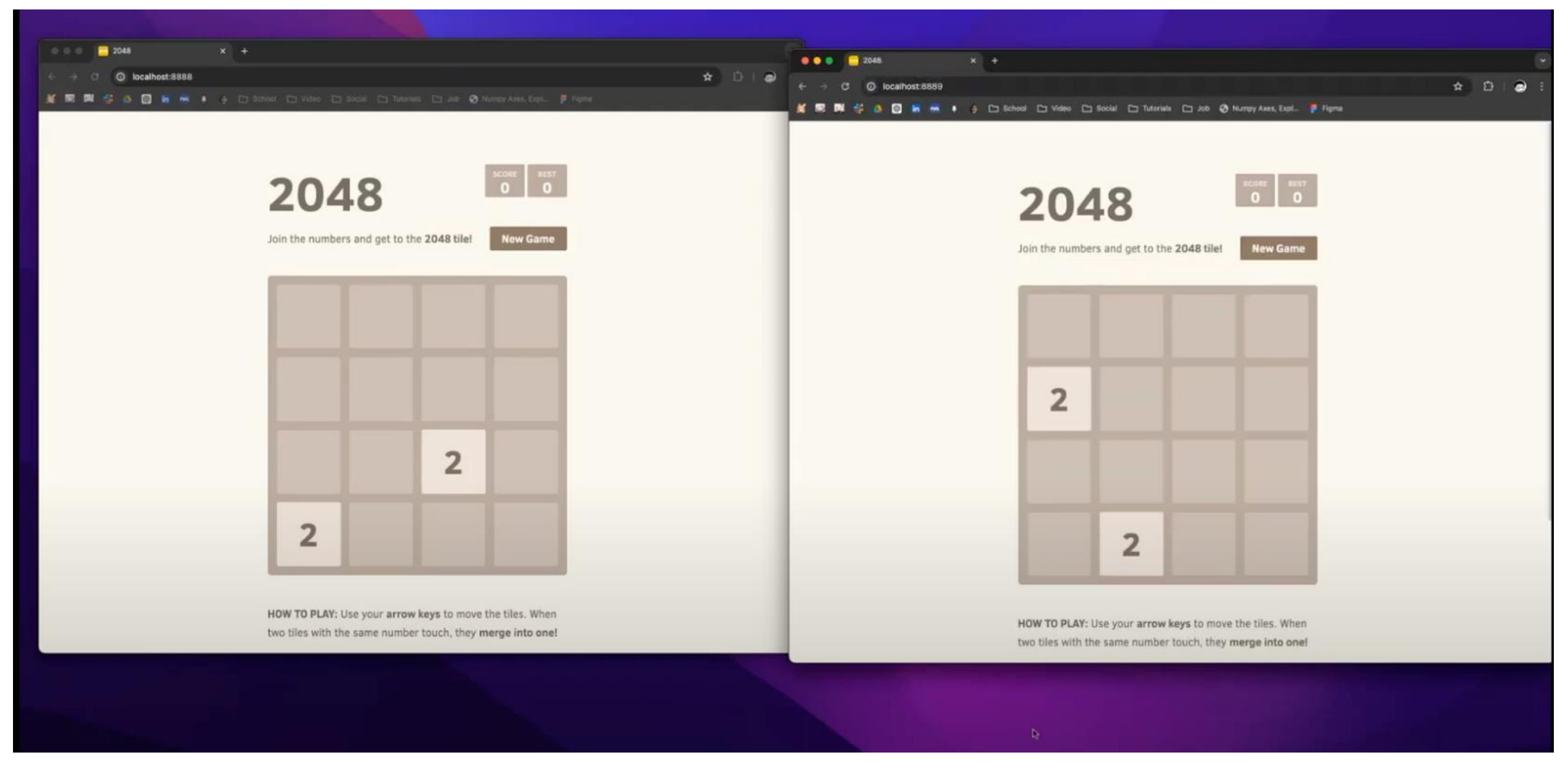


```
#!/usr/bin/env bash

set -e

echo -e "Checking that pre-req softwares are installed..."
curl -s -o check_pre_req.sh https://raw.githubusercontent.com/kubestellar/kubestellar/v0.23.1/hack/check_pre_req.sh
chmod +x check_pre_req.sh
./check_pre_req.sh -V

output=$(./check_pre_req.sh)

if echo "$output" | grep -q "X"; then
    echo "Please install all the necessary dependencies before contiuning"
    exit 1
else
    echo "Pre-req are properly installed. Proceeding to environment clean up"
fi

echo -e "\nStarting environment clean up..."
echo -e "Starting cluster clean up..."

cluster_clean_up() {
    error_message=$(eval "$1" 2>&1)
    if [ $? -ne 0 ]; then
        echo "clean up failed. Error:"
        echo "$error_message"
    fi
}

cluster_clean_up "kind delete cluster --name kubeflex"
cluster_clean_up "kind delete cluster --name cluster1"
cluster_clean_up "kind delete cluster --name cluster2"
echo -e "Cluster space clean up has been completed"

echo -e "\nStarting context clean up..."

context_clean_up() {
    output=$(kubectx)
    # echo $output
    while IFS= read -r line; do
        if [ "$line" == "kind-kubeflex" ]; then
            echo "Deleting kind-kubeflex context..."
            kubectl config delete-context kind-kubeflex
```

Bash Script

- 15+ lines of command to create 2 new remote clusters
- Create a script that cleans out old clusters and reactivate inactive pods
- Increase cluster creation efficiency by 25%
- Still takes about 5 minutes to create the clusters using this script

[Demo Link]



Future Work

- Fix/Critique KubeStellar's Documentation
- Improve script runtime to reduce cluster's creation time
- Fix port-forwarding issue with the clusters to deploy game-2048
- Work on windows version of the documentation on KubeStellar

Georgia Tech